




Centrum voor Wiskunde en Informatica

View metadata, citation and similar papers at core.ac.uk

brought to you by  **CORE**

provided by CWI's Institu

REPORTRAPPORT

MAS

Modelling, Analysis and Simulation



Modelling, Analysis and Simulation

Improving the parallel performance of a domain decomposition preconditioning technique in the Jacobi-Davidson method for large scale eigenvalue problems

M. Genseberger

REPORT MAS-R0805 AUGUST 2008

Centrum voor Wiskunde en Informatica (CWI) is the national research institute for Mathematics and Computer Science. It is sponsored by the Netherlands Organisation for Scientific Research (NWO). CWI is a founding member of ERCIM, the European Research Consortium for Informatics and Mathematics.

CWI's research has a theme-oriented structure and is grouped into four clusters. Listed below are the names of the clusters and in parentheses their acronyms.

Probability, Networks and Algorithms (PNA)

Software Engineering (SEN)

Modelling, Analysis and Simulation (MAS)

Information Systems (INS)

Copyright © 2008, Stichting Centrum voor Wiskunde en Informatica
P.O. Box 94079, 1090 GB Amsterdam (NL)
Kruislaan 413, 1098 SJ Amsterdam (NL)
Telephone +31 20 592 9333
Telefax +31 20 592 4199

ISSN 1386-3703

Improving the parallel performance of a domain decomposition preconditioning technique in the Jacobi-Davidson method for large scale eigenvalue problems

ABSTRACT

Most computational work in Jacobi-Davidson [9], an iterative method for large scale eigenvalue problems, is due to a so-called correction equation. In [5] a strategy for the approximate solution of the correction equation was proposed. This strategy is based on a domain decomposition preconditioning technique in order to reduce wall clock time and local memory requirements. This report discusses the aspect that the original strategy can be improved. For large scale eigenvalue problems that need a massively parallel treatment this aspect turns out to be nontrivial. The impact on the parallel performance will be shown by results of scaling experiments up to 1024 cores.

2000 Mathematics Subject Classification: 65F15; 65N25; 65F10; 65N55; 65Y05

Keywords and Phrases: eigenvalue problems; domain decomposition; Jacobi-Davidson; inexact Newton method; Schwarz method; Krylov method

Improving the parallel performance of a domain decomposition preconditioning technique in the Jacobi-Davidson method for large scale eigenvalue problems

Menno Genseberger

Deltares,

P.O. Box 177, 2600 MH Delft, The Netherlands

and

Environmental Fluid Mechanics Section, Faculty of Civil Engineering and Geosciences,

Delft University of Technology, P.O. Box 5048, 2600 GA Delft, The Netherlands

Menno.Genseberger@deltares.nl

July 31, 2008

Abstract

Most computational work in Jacobi-Davidson [9], an iterative method for large scale eigenvalue problems, is due to a so-called correction equation. In [5] a strategy for the approximate solution of the correction equation was proposed. This strategy is based on a domain decomposition preconditioning technique in order to reduce wall clock time and local memory requirements.

This report discusses the aspect that the original strategy can be improved. For large scale eigenvalue problems that need a massively parallel treatment this aspect turns out to be nontrivial. The impact on the parallel performance will be shown by results of scaling experiments up to 1024 cores.

2000 Mathematics Subject Classification: 65F15, 65N25, 65F10, 65N55, 65Y05

Keywords: eigenvalue problems, domain decomposition, Jacobi-Davidson, inexact Newton method, Schwarz method, Krylov method

1 Introduction

Eigenvalue problems show up in different scientific disciplines. For instance in modelling rotating plasma equilibria in tokamaks (fusion research) or in stability analysis of ocean circulation patterns (climatology/oceanography). Here, eigenmodes and eigenvalues of a specific partial differential operator describe characteristic properties of the underlying physical process. For simulation or stability analysis of such a physical process by computer, the corresponding partial differential operator is discretized on the domain of interest. For typical applications, this results in large scale (standard or generalized) eigenvalue problems.

The Jacobi-Davidson method [9] is an iterative method suitable for computing solutions of these kind of large scale eigenvalue problems. Most computational work of Jacobi-Davidson is due to the correction equation at the intermediate level. In [5, 6] a strategy for the computation of (approximate) solutions of the correction equation was proposed. The strategy is based on a domain decomposition technique [12, 13] in order to reduce wall clock time and local memory requirements: based on domain

decomposition both data and operations on this data can be distributed to enable parallel computing on distributed memory architectures. Because of the very large scales, for many applications, the eigenvalue problems also require such an approach.

Here, we discuss the aspect that the original strategy in [5, 6] can be improved by taking into account the relation of the intermediate level with the top level of the Jacobi-Davidson method. This results in a different application of the domain decomposition technique to the Jacobi-Davidson method. Although the two approaches look similar, there are subtle differences in implementation and the consequences in terms of computational time for large scale eigenvalue problems are nontrivial.

Note that, as Jacobi-Davidson can be viewed as an accelerated inexact Newton method [3], results of the research presented here may also be of interest to applications that use inexact Newton methods for large systems of nonlinear equations.

This report is organized as follows. First, in §2 we summarize the ingredients of the Jacobi-Davidson method. Then §3 outlines the domain decomposition preconditioning technique and its application to the correction equation. After this, we discuss in §4 the different levels in Jacobi-Davidson at which the domain decomposition preconditioning technique may be applied and motivate our choice for the top level. Finally, the practical consequences of the two approaches in terms of parallel performance are investigated by an extended set of scaling experiments in §5.

2 The Jacobi-Davidson method for eigenvalue problems

For ease of presentation, but without loss of generality, we consider the standard eigenvalue problem

$$\mathbf{A} \mathbf{x} = \lambda \mathbf{x} \quad (1)$$

for a simple eigenvalue λ and corresponding eigenvector \mathbf{x} of a square matrix \mathbf{A} .

As mentioned in §1, for our type of applications matrix \mathbf{A} originates from the discretization (by means of finite differences, elements, or volumes) of some partial differential operator.

For the eigenvalue problem (1) the Jacobi-Davidson method by Sleijpen and van der Vorst [9] computes iteratively a solution. The ingredients of each iteration are:

- **Extract** an approximate eigenpair $(\theta, \mathbf{u}) \approx (\lambda, \mathbf{x})$ from a search subspace \mathbf{V} via a *Rayleigh-Ritz* procedure.

The *Rayleigh* part projects \mathbf{A} on \mathbf{V} by constructing the *matrix Rayleigh quotient* of \mathbf{V} (or interaction matrix): $H \equiv \mathbf{V}^* \mathbf{A} \mathbf{V}$.

The *Ritz* part solves the projected eigenvalue problem $H \mathbf{s} = \theta \mathbf{s}$, selects the *Ritz value* θ closest to some target τ , and computes the corresponding *Ritz vector* $\mathbf{u} \equiv \mathbf{V} \mathbf{s}$ and residual $\mathbf{r} \equiv -(\mathbf{A} - \theta \mathbf{I}) \mathbf{u}$.

- **Correct** the approximate eigenvector \mathbf{u} .

The *correction vector* $\mathbf{t} \perp \mathbf{u}$ is computed from the *correction equation*

$$\mathbf{P} (\mathbf{A} - \theta \mathbf{I}) \mathbf{P} \mathbf{t} = \mathbf{r} \quad \text{with the projection} \quad \mathbf{P} \equiv \mathbf{I} - \frac{\mathbf{u} \mathbf{u}^*}{\mathbf{u}^* \mathbf{u}}. \quad (2)$$

- **Expand** the search subspace \mathbf{V} with the correction vector \mathbf{t} .

The Jacobi-Davidson method can be viewed as an accelerated (inexact) Newton method [3]. Here, the correction equation (2) describes a Newton step. It tries to get that characteristic information from the (large) eigenvalue problem (1) such that, after expansion of the subspace, from the (small) projected

eigenvalue problem a much better approximate eigenpair can be distilled. So, convergence of the method depends strongly on the accuracy of (approximate) solutions of the correction equation.

Most computational work of Jacobi-Davidson is also in the correction equation. In many practical applications exact solution of this equation is not feasible because of time and/or memory constraints. Then one has to rely on approximate solutions obtained from some iterative method for linear systems. For the convergence of such a method a good preconditioner is highly desirable, this will be the subject of the next section.

3 Domain decomposition technique for the correction equation

In [5, 6] a domain decomposition preconditioning technique for the (approximate) solution of the correction equation (2) was proposed. This technique is based on a nonoverlapping additive Schwarz method with locally optimized coupling parameters by Tan & Borsboom [12, 13] for ordinary linear systems which is a generalization of work by Tang [14]. The linear system that is described by the correction equation of the Jacobi-Davidson method, may be highly indefinite and is given in an unusual manner so that the application of the domain decomposition technique needed further development and special attention.

3.1 A locally optimized Schwarz method

We describe the domain decomposition technique for the two subdomain case. It can be generalized to more than two subdomains in a straightforward manner. Furthermore, for the discretization, instead of finite differences, as used here for ease of presentation, finite elements or volumes may be used.

Suppose some partial differential equation (PDE) is defined on a domain Ω with appropriate boundary conditions on the boundary $\partial\Omega$. In order to compute numerical solutions, Ω is covered by a grid $\hat{\Omega}$. The PDE is discretized accordingly, with unknowns defined on the grid points, yielding the linear system

$$\mathbf{B} \mathbf{y} = \mathbf{d}. \quad (3)$$

We decompose Ω in two nonoverlapping subdomains Ω_1 and Ω_2 with interface (or internal boundary) Γ . The subdomains are covered by subgrids $\hat{\Omega}_1$ and $\hat{\Omega}_2$. However, we require that no splitting of the original discretized operator (or stencil) has to be made. For that purpose, for each subdomain additional grid points are introduced, they are located just outside the subdomain near the interface Γ (the open bullets “o” in Fig. 1). Since this introduces extra unknowns on the additional grid points, we must also provide extra equations that describe these extra unknowns. Furthermore, for the exact solution of the discretized PDE we want the quantities on these additional points of one subgrid to be equal to the quantities on the grid points of the other subgrid on the same location. Now, the *enhancement* consists of providing the original system with extra unknowns at the additional grid points and extra equations with precisely this property.

For a typical discretization the matrix \mathbf{B} of (3) is banded and the unknowns are only locally coupled. Therefore, by proper ordering we can force that the linear system (3) has the following structure:

$$\begin{bmatrix} \mathbf{B}_{11} & \mathbf{B}_{1\ell} & \mathbf{0} & \mathbf{0} \\ \mathbf{B}_{\ell 1} & B_{\ell\ell} & B_{\ell r} & \mathbf{0} \\ \mathbf{0} & B_{r\ell} & B_{rr} & \mathbf{B}_{r2} \\ \mathbf{0} & \mathbf{0} & \mathbf{B}_{2r} & \mathbf{B}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{y}_1 \\ y_\ell \\ y_r \\ \mathbf{y}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{d}_1 \\ d_\ell \\ d_r \\ \mathbf{d}_2 \end{bmatrix}.$$

For \mathbf{B} , the labels 1, 2, ℓ , and r , respectively, refer to operations on data from/to subdomain Ω_1 , Ω_2 , and left, right from the interface Γ , respectively. For \mathbf{y} and \mathbf{d} , the labels 1, 2, ℓ , and r , respectively, refer

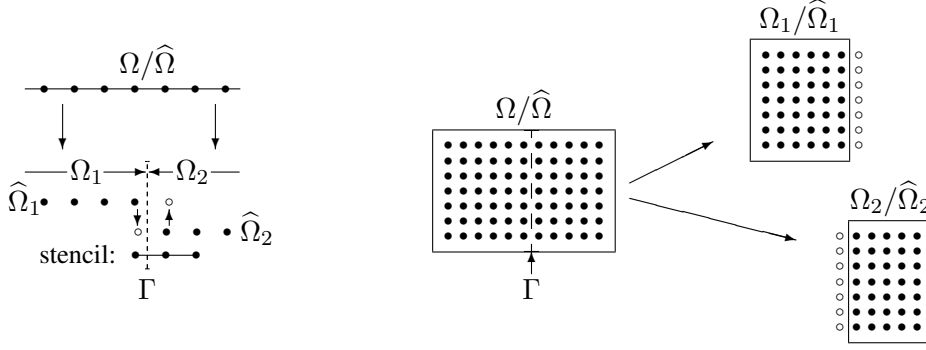


FIGURE 1. *Decomposition in one (left picture) and two dimensions (right picture).*

to data in subdomain Ω_1 , Ω_2 , and left, right from the interface Γ , respectively. Here, subvector y_ℓ (y_r respectively) contains those unknowns on the left (right) from Γ that are coupled by the stencil both with unknowns in Ω_1 (Ω_2) and unknowns on the right (left) from Γ . This explains the zeros in the expression for matrix \mathbf{B} .

We enhance the linear system (3) to $\mathbf{B}_C \mathbf{y} = \mathbf{d}_0$ with the following structure:

$$\begin{bmatrix} \boxed{\begin{matrix} \mathbf{B}_{11} & \mathbf{B}_{1\ell} & \mathbf{0} \\ \mathbf{B}_{\ell 1} & B_{\ell\ell} & B_{\ell r} \\ \mathbf{0} & C_{\ell\ell} & C_{\ell r} \end{matrix}} & \boxed{\begin{matrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ 0 & 0 & \mathbf{0} \\ -C_{\ell\ell} & -C_{\ell r} & \mathbf{0} \end{matrix}} \\ \boxed{\begin{matrix} \mathbf{0} & -C_{r\ell} & -C_{rr} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{matrix}} & \boxed{\begin{matrix} C_{r\ell} & C_{rr} & \mathbf{0} \\ B_{r\ell} & B_{rr} & \mathbf{B}_{r2} \\ \mathbf{0} & \mathbf{B}_{2r} & \mathbf{B}_{22} \end{matrix}} \end{bmatrix} \begin{bmatrix} \mathbf{y}_1 \\ y_\ell \\ \tilde{y}_r \\ \tilde{y}_\ell \\ y_r \\ \mathbf{y}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{d}_1 \\ d_\ell \\ 0 \\ 0 \\ d_r \\ \mathbf{d}_2 \end{bmatrix}. \quad (4)$$

Here \tilde{y}_r (\tilde{y}_ℓ respectively) contains the unknowns at the additional grid points (the open bullets “o” in Fig. 1) of the subgrid for Ω_1 (Ω_2) on the right (left) of Γ . We call \mathbf{B}_C the *enhancement* of \mathbf{B} , \mathbf{y} the *unbalanced enhancement* of \mathbf{y} , and \mathbf{d}_0 the *zero enhancement* of \mathbf{d} . For the special case that $\tilde{y}_\ell = y_\ell$ and $\tilde{y}_r = y_r$ we introduce also the *balanced enhancement* \mathbf{y} of \mathbf{y} .

The only requirement for the extra equations in (4) is that the submatrix (the *interface coupling matrix*)

$$C \equiv \begin{bmatrix} C_{\ell\ell} & C_{\ell r} \\ C_{r\ell} & C_{rr} \end{bmatrix}$$

is nonsingular. For nonsingular C the solution of the enhanced system (4) is unique, $\tilde{y}_\ell = y_\ell$ and $\tilde{y}_r = y_r$ as required (i.e. $\mathbf{y} = \mathbf{y}$), and the restriction of this solution \mathbf{y} to \mathbf{y} is the unique solution of the original linear system (3) ([12, Theorem 1], [13, Theorem 1.2.1]).

However, we want to perform solves on the subgrids only. For that purpose we split the matrix of the enhanced system (4) as $\mathbf{B}_C = \mathbf{M} - \mathbf{N}$. Here \mathbf{M} is composed of the boxed parts in (4) that do not map elements from one subgrid to the other subgrid. Note that the remainder \mathbf{N} has a relatively small number of nonzero elements. (Therefore matrix vector multiplication with \mathbf{B}_C can be implemented efficiently on distributed memory computers by means of this domain decomposition technique.) For the splitting a simple iterative solution method is the Richardson iteration:

$$\mathbf{y}^{(i+1)} = \mathbf{y}^{(i)} + \mathbf{M}^{-1} (\mathbf{d}_0 - \mathbf{B}_C \mathbf{y}^{(i)}). \quad (5)$$

In practice, for instance on a distributed memory computer, the action of \mathbf{M}^{-1} can be implemented by performing solves for each boxed part in (4) separately. This enables simultaneous solves on all subgrids:

that is precisely the situation that we want for massively parallel computing. But, the subproblems on the subgrids are still coupled by \mathbf{N} . By rewriting (5) one observes that the contribution of \mathbf{N} , which is neglected by the subdomain solves, is corrected by an explicit step (which requires data exchange between subdomains):

$$\text{solve } \mathbf{y}^{(i+1)} \text{ from } \mathbf{M} \mathbf{y}^{(i+1)} = \mathbf{d}_0 + \mathbf{N} \mathbf{y}^{(i)}.$$

Because of this, due to the splitting the iterates $\mathbf{y}^{(i)}$ are perturbed by errors. Each step these errors are amplified by the *error propagation matrix* $\mathbf{M}^{-1} \mathbf{N}$.

Now, the key idea is to use the degrees of freedom, that we have created by the introduction of additional unknowns near the interface, for damping the error components. For this purpose, the spectral properties of $\mathbf{M}^{-1} \mathbf{N}$ for the specific underlying PDE are analyzed. With results of this analysis, optimal coupling parameters can be estimated, i.e. the interface coupling matrix \mathbf{C} can be tuned.

In this way error components due to the splitting are damped “as much as possible”, optimal choices result in a coupling that annihilates the outflow from one domain to another: absorbing boundary conditions. This leads effectively to almost uncoupled subproblems at subdomains. As a consequence, the number of iterations of (5) required for convergence is minimal with minimal communication overhead (due to the explicit step with \mathbf{N}) between subdomains: an ideal situation for implementation on parallel computers and/or distributed memory.

For ordinary systems of linear equations originating from advection dominated problems tuning of the interface coupling matrix was done in [12, 13]. For the correction equation originating from diffusion dominated eigenvalue problems this was done in [5, 6], the main results will be summarized in the next subsection.

Besides the tuning of the interface coupling matrix we can further speed up the process. The Richardson iteration uses only information from the last iterate for the computation of a new one. The process can be accelerated by collecting the iterates in the Krylov subspace

$$\mathcal{K}_m = \text{span} \left(\mathbf{M}^{-1} \mathbf{d}_0, \mathbf{M}^{-1} \mathbf{B}_C \mathbf{M}^{-1} \mathbf{d}_0, \dots, (\mathbf{M}^{-1} \mathbf{B}_C)^{m-1} \mathbf{M}^{-1} \mathbf{d}_0 \right)$$

and computing an approximate solution for (4) with respect to \mathcal{K}_m . In fact, in this way the Krylov method computes a solution for the *left* preconditioned equation $\mathbf{M}^{-1} \mathbf{B}_C \mathbf{y} = \mathbf{M}^{-1} \mathbf{d}_0$. Note that *right* preconditioning is possible as well and has the nice additional property that for exact solves with \mathbf{M} the linear system can be transformed into a much smaller linear system for unknowns near the interface Γ (Schur complement, see [6, §3.2.4]). This is of interest as for large scale eigenvalue problems the corresponding correction equation has the same, large scale, dimensions.

3.2 Situation for the correction equation

Here, first it is shown how the correction equation is enhanced and how the preconditioner is incorporated. Then we pay attention to the spectrum of the error propagation matrix for a model eigenvalue problem and outline, with this knowledge in mind, the strategy for the tuning of the interface coupling matrix.

Similar to the enhancements (4) of the linear system (3) in the previous subsection, the following components of the correction equation are enhanced: the matrix $\mathbf{B} \equiv \mathbf{A} - \theta \mathbf{I}$ to \mathbf{B}_C , the correction vector \mathbf{t} to $\tilde{\mathbf{t}}$ (unbalanced enhancement), and the vectors \mathbf{u} and \mathbf{r} to \mathbf{u}_0 and \mathbf{r}_0 , respectively (zero enhancements). With these enhancements, a correction $\tilde{\mathbf{t}} \perp \mathbf{u}_0$ is computed from the following enhanced correction equation [6, §3.3.2]:

$$\mathbf{P} \mathbf{B}_C \mathbf{P} \tilde{\mathbf{t}} = \mathbf{r}_0 \quad \text{with} \quad \mathbf{P} \equiv \mathbf{I} - \frac{\mathbf{u}_0 \mathbf{u}_0^*}{\mathbf{u}_0^* \mathbf{u}_0}. \quad (6)$$

The preconditioner \mathbf{M} for \mathbf{B}_C is constructed in the same way as the ordinary linear system case shown by the boxed parts in (4). However, because of the indefiniteness, for the correction equation the matrices \mathbf{B}_C and \mathbf{M} are accompanied by projections. Both for left and right preconditioning the projection is as follows:

$$\mathbf{P}' \equiv \mathbf{I} - \frac{\mathbf{M}^{-1} \mathbf{u}_0 \mathbf{u}_0^*}{\mathbf{u}_0^* \mathbf{M}^{-1} \mathbf{u}_0}. \quad (7)$$

In case of left preconditioning (for right preconditioning see [6, §3.3.3]) we compute approximate solutions to the correction equation from

$$\mathbf{P}' \mathbf{M}^{-1} \mathbf{B}_C \mathbf{P}' \tilde{\mathbf{t}} = \mathbf{P}' \mathbf{M}^{-1} \mathbf{r}_0. \quad (8)$$

In [5] and [6, §3.4.3] the spectral properties of the error propagation matrix $\mathbf{M}^{-1} \mathbf{N}$ were analyzed for the eigenvalue problem of an advection-diffusion operator with constant coefficients on two subdomains. In this analysis, the discretized operator was written as a tensor product of one-dimensional discretized advection diffusion operators. It turned out that, for the correction equation, the eigenvectors of the error propagation matrix show two typical types of behavior. Parallel to the interface Γ , all eigenvectors are coupled by eigenvectors of the one-dimensional discretized operator in the y -direction. Because of this, for the subblocks $C_{\ell\ell}$, $C_{\ell r}$, $C_{r\ell}$, and C_{rr} of the interface coupling matrix C we can take any linear combination of powers of the one-dimensional discretized operator in the y -direction. This introduces some freedom reflected by a number of coupling parameters. We can use these parameters for the minimization of the spectral radius of the error propagation matrix.

Perpendicular to the interface, however, there are differences. Most of the eigenvectors of the error propagation matrix show exponential behavior in the x -direction, the error grows exponentially fast when moving towards the interface Γ . For these eigenvectors we can estimate effective coupling parameters without specific knowledge of the subdomain size. This is of interest for practical situations with more than two subdomains, complicated geometry [6, §4.4] and variable coefficients [6, §4.3]. For this reason we minimize the spectral radius of the error propagation matrix only with respect to these eigenvectors.

A small number (this number grows when $\mathbf{B} \equiv \mathbf{A} - \theta \mathbf{I}$ and therefore \mathbf{B}_C becomes more indefinite, i.e. when θ is shifted more to the interior of the spectrum of \mathbf{A}) shows harmonic behavior in the x -direction, which has the disadvantage of being global. These eigenvectors are controlled with information from the search subspace of Jacobi-Davidson itself, as we will shortly describe now. In practice one starts the computations of Jacobi-Davidson with the extremal eigenvalues and once detected moves on to the less extremal ones (θ is shifted more to the interior of the spectrum of \mathbf{A}). Now, the eigenvectors corresponding to the extremal eigenvalues that are already computed contain precisely those harmonic components in the x -direction. In the Jacobi-Davidson method these eigenvectors are deflated. As a consequence deflation helps to cluster the part of the spectrum that we do not control with the coupling parameters, and therefore improves the conditioning of the preconditioned correction equation.

4 Different levels suitable in the Jacobi-Davidson method

However, given the domain decomposition preconditioning technique from the previous section, there is more to gain. For approximate solves of the correction equation with a preconditioned Krylov method, the Jacobi-Davidson method is an accelerated inexact Newton method [3] that consists of two nested iterative solvers. In the innerloop a search subspace for the (approximate) solution of the correction equation is built up by powers of the preconditioned matrix $\mathbf{A} - \theta \mathbf{I}$ for fixed θ . In the outerloop a search subspace for the (approximate) solution of the eigenvalue problem is built up by powers of the preconditioned matrix $\mathbf{A} - \theta \mathbf{I}$ for variable θ . In [5, 6] the domain decomposition technique was applied

to the innerloop. But, as θ varies slightly in succeeding outer iterations, one may take advantage of the nesting by applying the same technique to the outerloop as we will discuss now.

4.1 Enhancement at different levels

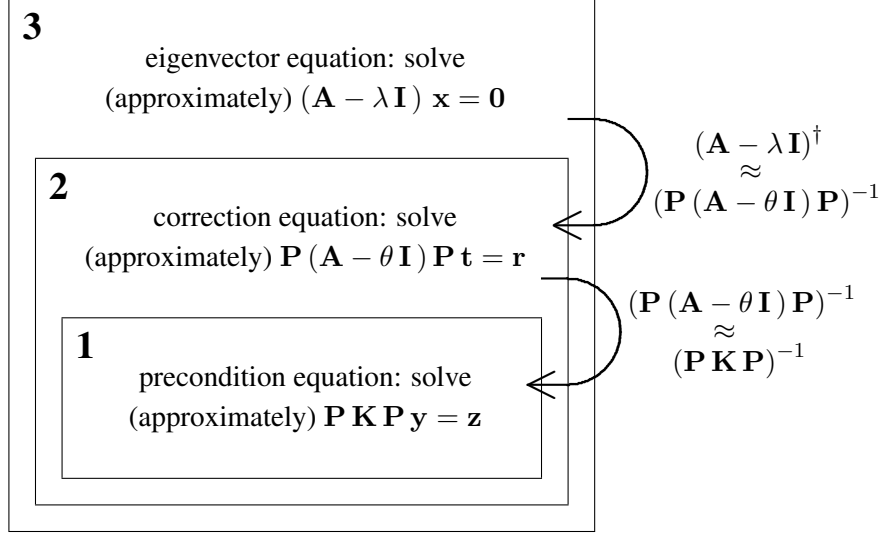


FIGURE 2. Three levels in the Jacobi-Davidson method suitable for enhancement.

Here we take a look at the levels in Jacobi-Davidson to which the domain decomposition preconditioning technique may be applied.

First note that, if an eigenvalue λ is known, then one can rewrite the eigenvalue problem (1) to the *eigenvector equation*

$$(\mathbf{A} - \lambda \mathbf{I}) \mathbf{x} = \mathbf{0} \quad (9)$$

which describes the corresponding eigenvector.

Three levels of the Jacobi-Davidson method are distinguished (indicated by numbers in Fig. 2): the eigenvector equation that describes the eigenvector on the highest level, the correction equation that describes the correction vector on the intermediate level, and the *precondition equation* that describes the preconditioned vector on the lowest level. The different levels are related by the involved linear operators (indicated by arrows in Fig. 2): as the exact eigenvalue λ in the operator $\mathbf{A} - \lambda \mathbf{I}$ is not known beforehand, it is replaced by an approximation θ , which leads to the operator $\mathbf{A} - \theta \mathbf{I}$. This operator is replaced by a preconditioner \mathbf{K} with which it is cheaper to solve systems. The projection \mathbf{P} , that accompanies the linear operators at level 1 and 2, is the result of approximating the space orthogonal to the (unknown) eigenvector \mathbf{x} (the space on which the pseudo inverse $(\mathbf{A} - \lambda \mathbf{I})^\dagger$ acts) with the space orthogonal to the (known) approximate eigenvector \mathbf{u} . The relationships between the levels are the motivation for the different levels of enhancement that will be considered now.

Enhancement of the precondition equation may only be of practical interest in case of inexact subdomain solves. At this lowest level, for exact subdomain solves, an effective \mathbf{M} requires knowledge of the value of the correction vector on the internal boundary Γ . As the correction vector is the unknown vector, that is not practical. Therefore enhancement of the precondition equation will not be considered furthermore.

The enhancement of the correction equation is given by (6).

With the notation from §3, we enhance the eigenvector equation (9) as follows:

$$(\mathbf{A} - \lambda \mathbf{I})_C \tilde{\mathbf{x}} = \mathbf{0}. \quad (10)$$

Here $\tilde{\mathbf{x}}$ is the unbalanced enhancement of the eigenvector \mathbf{x} . Now, we introduce the *zero enhancement* \mathbf{I}_0 of the identity matrix \mathbf{I} : this is the special case that \mathbf{I} is enhanced for a zero coupling matrix, i.e. $C = 0$. With this the enhanced eigenvector equation (10) can be rewritten into

$$(\mathbf{A}_C - \lambda \mathbf{I}_0) \tilde{\mathbf{x}} = \mathbf{0}.$$

The enhanced eigenvector equation yields a correction equation that differs from (6), this will be subject next.

4.2 Correction equation for the enhanced eigenvector equation

For the enhanced eigenvector equation (10), the corresponding correction equation has two subtle differences with the enhanced correction equation (6): a correction $\tilde{\mathbf{t}} \perp \mathbf{u}_0$ is computed from

$$\mathbf{P} (\mathbf{A}_C - \theta \mathbf{I}_0) \mathbf{P} \tilde{\mathbf{t}} = \mathbf{P} \tilde{\mathbf{r}} \quad \text{with} \quad \mathbf{P} \equiv \mathbf{I} - \frac{\mathbf{u}_0 \mathbf{u}_0^*}{\mathbf{u}_0^* \mathbf{u}_0} \quad (11)$$

and with residual $\tilde{\mathbf{r}} \equiv -(\mathbf{A}_C - \theta \mathbf{I}_0) \tilde{\mathbf{u}}$. Note that this residual also measures the errors $u_\ell - \tilde{u}_\ell$ and $u_r - \tilde{u}_r$.

Equation (11) is derived now. It is a straightforward “generalization” of the derivation of the correction equation for (1) via a first order correction approach [9]. Suppose we have computed a pair (θ, \mathbf{u}) that approximates some eigenpair (λ, \mathbf{x}) . Furthermore, also an unbalanced enhancement $\tilde{\mathbf{u}}$ of \mathbf{u} is available from the information collected so far. We want to compute a correction $\tilde{\mathbf{t}}$ to $\tilde{\mathbf{u}}$, such that $\tilde{\mathbf{x}} = \tilde{\mathbf{u}} + \tilde{\mathbf{t}}$. Here $\tilde{\mathbf{x}}$ is the balanced enhancement of \mathbf{x} . The enhanced eigenvector equation (10) yields

$$(\mathbf{A}_C - \lambda \mathbf{I}_0) (\tilde{\mathbf{u}} + \tilde{\mathbf{t}}) = \mathbf{0}.$$

We rewrite this as follows:

$$(\mathbf{A}_C - \theta \mathbf{I}_0) \tilde{\mathbf{t}} = \tilde{\mathbf{r}} + (\lambda - \theta) \mathbf{u}_0 + (\lambda - \theta) \mathbf{t}_0.$$

Here $\mathbf{u}_0 = \mathbf{I}_0 \tilde{\mathbf{u}}$ and $\mathbf{t}_0 = \mathbf{I}_0 \tilde{\mathbf{t}}$. For θ close to λ and $\tilde{\mathbf{u}}$ close to $\tilde{\mathbf{x}}$ the term $(\lambda - \theta) \mathbf{t}_0$ is of second order. This contribution is neglected:

$$(\mathbf{A}_C - \theta \mathbf{I}_0) \tilde{\mathbf{t}} = \tilde{\mathbf{r}} + (\lambda - \theta) \mathbf{u}_0. \quad (12)$$

The difference $\lambda - \theta$ on the right-hand side of (12) is not known. This contribution disappears by projecting on the space orthogonal to \mathbf{u}_0 :

$$\mathbf{P} (\mathbf{A}_C - \theta \mathbf{I}_0) \tilde{\mathbf{t}} = \mathbf{P} \tilde{\mathbf{r}} \quad \text{with} \quad \mathbf{P} \equiv \mathbf{I} - \frac{\mathbf{u}_0 \mathbf{u}_0^*}{\mathbf{u}_0^* \mathbf{u}_0}. \quad (13)$$

If convergence takes place, that is if θ converges to an eigenvalue λ , then the operator $\mathbf{A}_C - \theta \mathbf{I}_0$ in (13) becomes singular. Because of this we can not compute proper solutions for (13). We repair this by also restricting the domain of the operator $\mathbf{A}_C - \theta \mathbf{I}_0$ to the space orthogonal to \mathbf{u}_0 . Then, one arrives at (11), the correction equation for the enhanced eigenvector equation (10). Observe that on the right-hand side there is also a projection where in the correction equations (2) and (6) there is not. This is because $\tilde{\mathbf{r}}$ is not perpendicular to \mathbf{u}_0 in general.

4.3 Incorporation in the Jacobi-Davidson method

Now we will outline how the enhanced correction equation (6) and the correction equation for the enhanced eigenvector equation (11) are incorporated in the Jacobi-Davidson method. For approximate solves with the domain decomposition preconditioning technique this will effectively lead to two different processes as will be shown in the remainder of this report. Therefore, from now on, application of the domain decomposition technique to the intermediate level (i.e. enhancement of the correction equation) will be labeled with “Jacobi-Davidson with enhanced innerloop”, application to the highest level (i.e. enhancement of the eigenvector equation) with “Jacobi-Davidson with enhanced outerloop”.

For the incorporation we have to specify at which stage of Jacobi-Davidson matrices and vectors need to be enhanced and restricted:

Jacobi-Davidson with enhanced innerloop. Each outer iteration the correction equation (2) is enhanced.

For that purpose the operator $\mathbf{A} - \theta \mathbf{I}$ is enhanced to $\mathbf{A}_C - \theta \mathbf{I}_0$ and the vectors \mathbf{u} and \mathbf{r} to \mathbf{u}_0 and \mathbf{r}_0 respectively. It is easy to see that if the enhanced correction equation (6) is solved exactly then the solution is balanced: \mathbf{t} and the restriction \mathbf{t} of this \mathbf{t} is also the unique solution of the original correction equation (2). However, if the enhanced correction equation (6) is solved only approximately then the solution $\tilde{\mathbf{t}}$ is unbalanced in general. For the next outer iteration we restrict $\tilde{\mathbf{t}}$ to \mathbf{t} and expand \mathbf{V} with this \mathbf{t} . Jacobi-Davidson continues with the Rayleigh-Ritz procedure of §2.

Jacobi-Davidson with enhanced outerloop. The vectors on the highest level, the level of the eigenvector equation, are enhanced. During the Jacobi-Davidson outer iterations an enhanced subspace $\tilde{\mathbf{V}}$ is built up. By means of the Rayleigh-Ritz procedure, a new approximate eigenpair (θ, \mathbf{u}) for the original eigenvalue problem (1) is computed with respect to the restricted subspace \mathbf{V} of $\tilde{\mathbf{V}}$. The enhanced vector $\tilde{\mathbf{u}} \equiv \tilde{\mathbf{V}} s$ corresponding to $\mathbf{u} \equiv \mathbf{V} s$ with residual $\tilde{\mathbf{r}} \equiv -(\mathbf{A}_C - \theta \mathbf{I}_0) \tilde{\mathbf{u}}$ is formed. We take this unbalanced enhancement $\tilde{\mathbf{u}}$ as it contains more information than the balanced enhancement \mathbf{u} . For approximate solutions of the correction equation (11) this will turn out to be more efficient for the overall process (see §4.4 and §5).

For $\tilde{\mathbf{u}} = \mathbf{u}$ the residual $\tilde{\mathbf{r}}$ equals \mathbf{r}_0 and is perpendicular to \mathbf{u}_0 . For this special case it is easy to see that the correction equations (11) and (6) are identical. When solved exactly both correction equations yield the same balanced correction vector \mathbf{t} .

In general the solution of (11) is unbalanced. As new approximate solutions to the original eigenvalue problem (1) are extracted from \mathbf{V} we need an orthonormal $\tilde{\mathbf{V}}$. Therefore we orthonormalize $\tilde{\mathbf{t}}$ with respect to the semi inner product defined by

$$\langle \mathbf{y}, \mathbf{z} \rangle \equiv \mathbf{y}^* \mathbf{I}_0 \mathbf{z} \quad \text{for } \mathbf{y}, \mathbf{z} \in \tilde{\mathbf{V}}. \quad (14)$$

Then $\tilde{\mathbf{V}}$ is expanded.

Overall we conclude the following: if the two processes are started with the same search subspace \mathbf{V} which is enhanced to a balanced $\tilde{\mathbf{V}}$ for the enhanced outerloop and if the correction equations are solved exactly, then the two processes are equivalent.

This conclusion is of theoretical interest only. The enhancements are introduced in order to accomodate a preconditioner based on domain decomposition. In practice, approximate solutions to the correction equation are computed by means of such a preconditioner \mathbf{M} . Note that as $\mathbf{B}_C \equiv (\mathbf{A} - \theta \mathbf{I})_C = \mathbf{A}_C - \theta \mathbf{I}_0$, the same $\mathbf{M} \approx \mathbf{B}_C$ can be used to precondition both correction equation (6) and correction equation (11). The next subsection discusses the effect of this preconditioning on the two overall processes.

4.4 Preconditioning: different processes

Both processes, Jacobi-Davidson with enhanced innerloop and enhanced outerloop, compute an approximate solution for the correction equation in an enhanced subspace built by powers of $\mathbf{M}^{-1} (\mathbf{A}_C - \theta \mathbf{I}_0)$ times a vector. For the enhanced correction equation it was shown in §3 how to incorporate a preconditioner. In order to accomodate also for an unbalanced $\tilde{\mathbf{u}}$, we will discuss here how a preconditioner can be incorporated in correction equation (11). Similar to [9, §4.1] first a one-step approximation is considered. This makes it easier to emphasize the difference by means of an example. It also facilitates the interpretation of the approximate solution of (11) with a preconditioned Krylov method later on.

4.4.1 One-step approximation

For Jacobi-Davidson with enhanced innerloop the one-step approximation is given by (cf. step 1 in [5, §3.3.3]):

$$\tilde{\mathbf{t}}^{(0)} = \mathbf{P}' \mathbf{M}^{-1} \mathbf{r}_0 \quad \text{with} \quad \mathbf{P}' \equiv \mathbf{I} - \frac{\mathbf{M}^{-1} \mathbf{u}_0 \mathbf{u}_0^*}{\mathbf{u}_0^* \mathbf{M}^{-1} \mathbf{u}_0}. \quad (15)$$

By premultiplying from the left with a preconditioner $\mathbf{M} \approx \mathbf{A}_C - \theta \mathbf{I}_0$ and imposing that the approximate correction vector is orthogonal to \mathbf{u}_0 , equation (12) yields a one-step approximation for Jacobi-Davidson with enhanced outerloop:

$$\tilde{\mathbf{t}}^{(0)} = \mathbf{P}' \mathbf{M}^{-1} \tilde{\mathbf{r}} \quad \text{with} \quad \mathbf{P}' \equiv \mathbf{I} - \frac{\mathbf{M}^{-1} \mathbf{u}_0 \mathbf{u}_0^*}{\mathbf{u}_0^* \mathbf{M}^{-1} \mathbf{u}_0}. \quad (16)$$

The only difference between (15) and (16) is the residual: for $\tilde{\mathbf{u}} \neq \underline{\mathbf{u}}$ the residuals $\tilde{\mathbf{r}}$ and \mathbf{r}_0 are not equal. In that case the solutions of (15) and (16) may differ. The appearance of an unbalanced $\tilde{\mathbf{u}} \neq \underline{\mathbf{u}}$ in Jacobi-Davidson with enhanced outerloop is very likely when the correction equation is not solved exactly. This is illustrated by the example that follows next. It also shows why it may be attractive to allow for an unbalanced Jacobi-Davidson search subspace.

4.4.2 Example

Suppose that Jacobi-Davidson with enhanced innerloop is started with $\mathbf{V}^{(0)} \equiv \mathbf{u}^{(0)}$ and Jacobi-Davidson with enhanced outerloop with $\tilde{\mathbf{V}}^{(0)} \equiv \underline{\mathbf{u}}^{(0)}$, where $\underline{\mathbf{u}}^{(0)}$ is the balanced enhancement $\underline{\mathbf{u}}^{(0)}$ of $\mathbf{u}^{(0)}$. Then we have that

$$\theta_0 = \frac{(\mathbf{u}^{(0)})^* \mathbf{A} \mathbf{u}^{(0)}}{(\mathbf{u}^{(0)})^* \mathbf{u}^{(0)}}, \quad (17)$$

for both processes. For simplicity of notation, let $\mathbf{B}_{C_0} = \mathbf{A}_{C_0} - \theta_0 \mathbf{I}_0$ with coupling C_0 . Given some preconditioner $\mathbf{M}_{C_0} \approx \mathbf{B}_{C_0}$, the one-step approximations (15) and (16) yield the same approximate correction vector $\tilde{\mathbf{t}}^{(0)} = -\mathbf{P}^{(0)} \mathbf{M}_{C_0}^{-1} \mathbf{B}_{C_0} \underline{\mathbf{u}}^{(0)}$, with

$$\mathbf{P}^{(0)} \equiv \mathbf{I} - \frac{\mathbf{M}_{C_0}^{-1} \mathbf{u}_0^{(0)} (\mathbf{u}_0^{(0)})^*}{(\mathbf{u}_0^{(0)})^* \mathbf{M}_{C_0}^{-1} \mathbf{u}_0^{(0)}}.$$

Note that in general $\tilde{\mathbf{t}}^{(0)}$ is unbalanced. Jacobi-Davidson with enhanced innerloop deals with this unbalanced vector $\tilde{\mathbf{t}}^{(0)}$ by restricting it to $\mathbf{t}^{(0)}$. With this vector the search subspace is expanded to $\mathbf{V}^{(1)} \equiv \text{span}(\mathbf{u}^{(0)}, \mathbf{t}^{(0)})$. The new search subspace $\tilde{\mathbf{V}}^{(1)}$ of Jacobi-Davidson with enhanced outerloop is spanned by $\underline{\mathbf{u}}^{(0)}$ and $\tilde{\mathbf{t}}^{(0)}$.

The next outer iteration a new approximate eigenpair is determined. As the restriction of $\mathbf{V}^{(1)}$ is equal to $\mathbf{V}^{(1)}$, the interaction matrices of the two processes are identical. Because of this, both processes determine the same new approximate eigenvalue, the Ritz value θ_1 . Let the corresponding Ritz vector be $\mathbf{u}^{(1)}$. Jacobi-Davidson with enhanced innerloop enhances this vector into a balanced vector $\underline{\mathbf{u}}^{(1)}$ that can be written as

$$\underline{\mathbf{u}}^{(1)} = \alpha \underline{\mathbf{u}}^{(0)} + \beta \underline{\mathbf{t}}^{(0)}$$

for some α and β . These coefficients α and β also describe the unbalanced enhanced Ritz vector $\tilde{\mathbf{u}}^{(1)}$ of Jacobi-Davidson with enhanced outerloop:

$$\tilde{\mathbf{u}}^{(1)} = \alpha \underline{\mathbf{u}}^{(0)} + \beta \tilde{\mathbf{t}}^{(0)},$$

so, for this case, an approximate solution of the correction equation leads to an unbalanced $\tilde{\mathbf{u}} \neq \underline{\mathbf{u}}$.

If one proceeds, Jacobi-Davidson with enhanced innerloop computes a new approximate correction vector equal to

$$\mathbf{P}^{(1)} \mathbf{M}_{C_1}^{-1} \mathbf{B}_{C_1} \underline{\mathbf{u}}^{(1)} = \alpha \mathbf{P}^{(1)} \mathbf{M}_{C_1}^{-1} \mathbf{B}_{C_1} \underline{\mathbf{u}}^{(0)} + \beta \mathbf{P}^{(1)} \mathbf{M}_{C_1}^{-1} \mathbf{B}_{C_1} \underline{\mathbf{t}}^{(0)}. \quad (18)$$

However, the new approximate correction vector for Jacobi-Davidson with enhanced outerloop can be written as an operator applied to the start vector $\underline{\mathbf{u}}^{(0)}$:

$$\begin{aligned} \mathbf{P}^{(1)} \mathbf{M}_{C_1}^{-1} \mathbf{B}_{C_1} \tilde{\mathbf{u}}^{(1)} &= \alpha \mathbf{P}^{(1)} \mathbf{M}_{C_1}^{-1} \mathbf{B}_{C_1} \underline{\mathbf{u}}^{(0)} + \beta \mathbf{P}^{(1)} \mathbf{M}_{C_1}^{-1} \mathbf{B}_{C_1} \tilde{\mathbf{t}}^{(0)} \\ &= \left(\alpha \mathbf{P}^{(1)} \mathbf{M}_{C_1}^{-1} \mathbf{B}_{C_1} + \beta \mathbf{P}^{(1)} \mathbf{M}_{C_1}^{-1} \mathbf{B}_{C_1} \mathbf{P}^{(0)} \mathbf{M}_{C_0}^{-1} \mathbf{B}_{C_0} \right) \underline{\mathbf{u}}^{(0)}. \end{aligned} \quad (19)$$

We consider a coupling C_i that is tuned as in §3. As already remarked in §3, such a coupling damps out errors by increasing powers of $\mathbf{M}_{C_i}^{-1} \mathbf{B}_{C_i}$. Furthermore, if θ_1 is close to θ_0 , which is the case when Jacobi-Davidson is in the region of convergence, then for the optimized coupling, C_1 is close to C_0 and, as a result, $\mathbf{B}_{C_1} \approx \mathbf{B}_{C_0}$ and $\mathbf{M}_{C_1} \approx \mathbf{M}_{C_0}$. Because of this, in equation (19) remaining error components from the previous outer iteration are damped in the next outer iteration. In equation (18), however, the damping of these error components is disturbed in the next outer iteration.

From this example we learn that, if approximate solutions for the correction equation are obtained from a one-step approximation, then we may expect that Jacobi-Davidson with enhanced outerloop converges faster than Jacobi-Davidson with enhanced innerloop.

4.4.3 Higher order approximations

From §4.3 we know that for exact solutions of the correction equation Jacobi-Davidson with enhanced innerloop and Jacobi-Davidson with enhanced outerloop are equivalent. In the previous example we observed that for one-step approximations Jacobi-Davidson with enhanced outerloop will probably converge faster than Jacobi-Davidson with enhanced innerloop. The question remains how the two processes are related for higher order approximate solutions of the correction equation.

Jacobi-Davidson with enhanced innerloop computes such a solution with a preconditioned Krylov method. For that purpose equation (8) of §3 shows how to incorporate a preconditioner in the correction equation (6).

The situation for Jacobi-Davidson with enhanced outerloop is considered now. A higher order approximation for a solution of (11) can be obtained by considering not only $\tilde{\mathbf{t}}^{(0)}$ from (16) but also terms defined by the sequence

$$\tilde{\mathbf{z}}^{(i)} = \mathbf{P}' \mathbf{M}^{-1} \mathbf{B}_C \tilde{\mathbf{z}}^{(i-1)} \quad \text{with} \quad \mathbf{P}' \equiv \mathbf{I} - \frac{\mathbf{M}^{-1} \mathbf{u}_0 \mathbf{u}_0^*}{\mathbf{u}_0^* \mathbf{M}^{-1} \mathbf{u}_0},$$

$\mathbf{B}_C = \mathbf{A}_C - \theta \mathbf{I}_0$, and $\mathbf{z}^{(0)} = \mathbf{t}^{(0)}$ for $i = 1, 2, \dots, m-1$ (cf. [10, §3.1.1]). These vectors $\mathbf{z}^{(i)}$ span the Krylov subspace $\mathcal{K}_m(\mathbf{P}' \mathbf{M}^{-1} \mathbf{B}_C, \mathbf{P}' \mathbf{M}^{-1} \mathbf{r})$. Note again, that the coupling C in §3 is chosen such that (most) error components induced by the splitting $\mathbf{B}_C = \mathbf{M} - \mathbf{N}$ are damped out by increasing powers of $\mathbf{M}^{-1} \mathbf{B}_C$. So for larger m , a better approximation $\mathbf{t}^{(m)}$ to the solution of (11) can be extracted from the Krylov subspace, for instance, with GMRES which computes the solution in \mathcal{K}_m that has a minimal residual in ℓ_2 -norm [7].

In fact, in this way, with a Krylov method an approximate solution $\mathbf{t}^{(m)}$ is computed to the preconditioned correction equation

$$\mathbf{P}' \mathbf{M}^{-1} \mathbf{B}_C \mathbf{P}' \mathbf{t} = \mathbf{P}' \mathbf{M}^{-1} \mathbf{r} \quad \text{with} \quad \mathbf{P}' \equiv \mathbf{I} - \frac{\mathbf{M}^{-1} \mathbf{u}_0 \mathbf{u}_0^*}{\mathbf{u}_0^* \mathbf{M}^{-1} \mathbf{u}_0}. \quad (20)$$

Again, it can be shown that for $\mathbf{u} = \mathbf{u}$ the preconditioned correction equations (8) and (20) are identical.

As for higher order solutions of the correction equations (8) and (20) the error components are damped more in one single innerloop, it is to be expected that the difference between the two processes, as illustrated in the example, becomes less significant: damping due to the outerloop in the process with enhanced eigenvector equation then has a smaller contribution to the overall process. This last expectation is verified numerically by means of a simple MATLAB experiment.

4.4.4 Simple MATLAB experiment

For this simple experiment we consider the eigenvalue problem for the Laplace operator defined on $\Omega = [0, 1]^2$. The domain Ω is decomposed into 8×8 square subdomains. With central differences we discretize the Laplace operator on Ω . Each subdomain is covered by a 25×25 subgrid as explained in §3.1. For this discretized problem, with Jacobi-Davidson we compute the most global eigenvector. We start Jacobi-Davidson with a parabolic shaped vector (see footnote in §5). Approximate solutions to the correction equations (8) and (20) in the two processes are solved by right preconditioned GMRES(m) for exact solves with the preconditioner \mathbf{M} . The number of GMRES-steps m is kept fixed for each outer iteration. As the difference of these processes is expected to depend on the accuracy of the approximate solution of the correction equation, three values of m are considered: $m = 4, 8$, and 16 .

Table 1 shows the convergence history of the two processes. For each process we have listed: the error in the eigenvalue (columns 2 and 4) and the ℓ_2 -norm of the (restricted) residual (columns 3 and 5). For all three values of m the error in the eigenvalue at step 2 shows no difference for both processes. This is explained by example §4.4.2: because of the startvector both processes compute the same correction vector \mathbf{t} , this results in the same interaction matrix at step 2 from which the same approximate eigenvalue θ is determined. Note that for this argument the accuracy of \mathbf{t} does not matter, the point is that \mathbf{t} is not different here for the two processes. However, at step 2, the values of the residual norm differ for $m = 4$ and $m = 8$. For $m = 4$ the value is about 50 times smaller for Jacobi-Davidson with enhanced outerloop than Jacobi-Davidson with enhanced innerloop, for $m = 8$ about 40 times, and for $m = 16$ no difference can be observed. So for a more accurate correction vector the difference diminishes, as anticipated in §4.4.3

After step 2 also the approximate eigenvalue is different for the two processes. From the results in the table one observes that if the correction equations are solved with GMRES(4), that is solutions are of low accuracy, then Jacobi-Davidson with enhanced innerloop indeed needs significantly more outer iterations than Jacobi-Davidson with enhanced outerloop for convergence.

For this simple MATLAB example, we checked that, for the same number of outer iterations, both processes need nearly the same amount of flops. However, we are interested what the consequences of these observed trends are, in terms of computational time, for large scale eigenvalue problems. Therefore, we end with a number of scaling experiments on parallel computers in the next section.

		enhanced innerloop		enhanced outerloop	
step		$\theta - \lambda$	$\ \mathbf{r}\ _2$	$\theta - \lambda$	$\ \mathbf{r}\ _2$
GMRES(4)					
1		-2.61e-01	6.23e+00	-2.61e-01	6.23e+00
2		-4.87e-03	1.14e+01	-4.87e-03	2.23e-01
3		-2.24e-04	4.00e+00	-1.94e-05	1.13e-01
4		-1.49e-06	3.19e-01	-1.96e-08	1.13e-04
5		-3.85e-08	4.24e-02	-3.75e-11	6.59e-05
6		-1.34e-09	8.59e-03	-9.38e-13	1.21e-06
7		-3.35e-11	1.67e-03	-7.07e-13	1.34e-07
8		-1.22e-12	2.04e-04	-1.95e-13	4.43e-09
9		-2.03e-13	2.23e-05	-9.91e-13	1.47e-10
10		2.72e-12	2.66e-06		
11		1.49e-12	2.42e-07		
12		-6.44e-12	3.23e-08		
13		1.84e-12	2.70e-09		
14		-6.48e-12	3.42e-10		
GMRES(8)					
1		-2.61e-01	6.23e+00	-2.61e-01	6.23e+00
2		-5.16e-06	4.67e-01	-5.16e-06	1.12e-02
3		-6.00e-10	6.65e-03	-8.31e-11	1.59e-04
4		-2.70e-13	9.88e-05	-6.11e-13	4.78e-07
5		-2.59e-13	2.43e-06	-4.83e-13	5.01e-10
6		1.99e-13	1.61e-08		
7		-3.87e-13	3.39e-10		
GMRES(16)					
1		-2.61e-01	6.23e+00	-2.61e-01	6.23e+00
2		-2.22e-07	1.13e-02	-2.22e-07	1.13e-02
3		-5.33e-14	9.41e-07	-6.18e-13	2.41e-09
4		-5.33e-14	1.10e-10	-5.83e-13	2.70e-11

TABLE 1. *Dependency convergence behavior of two Jacobi-Davidson processes on accurate (approximate) correction vector.*

5 Scaling experiments

In this section results are presented of scaling experiments to investigate the parallel performance of Jacobi-Davidson with enhanced innerloop and Jacobi-Davidson with enhanced outerloop for large scale eigenvalue problems.

We start with investigating the behavior of the two processes for:

- different eigenvalue problems (§5.1),
- different stopping strategies for the inner solver (§5.2), and
- different inner solvers (§5.3).

Furthermore the speed-up is considered in (§5.4). These experiments are performed for eigenvalue problems of “intermediate” size (from 0.26 million up to 4 million unknowns) on the DAS-2 linux-cluster (from 4 up to 64 nodes, two 1-Ghz Intel Pentium-III cores and 1 GB shared memory per node, Myrinet-2000 network with 400 MB/s bandwidth and $3\mu\text{s}$ latency between nodes).

Then, in §5.5 we study the massively parallel behavior of the two processes. Scaling experiments are performed for the eigenvalue problem of the Laplace operator up to 67 million unknowns on three different supercomputers:

- the DAS-3 linux-cluster (two 2.4-Ghz AMD Opteron cores and 4 GB shared memory per node, Myri-10G network with 1.2-2.4 GB/s bandwidth and $2.2\mu\text{s}$ latency between nodes),
- the LISA linux-cluster (two 3.4-Ghz Intel Xeon cores and 2 GB shared memory per node, Infini-Band network with 800 MB/s bandwidth and $6\mu\text{s}$ latency between nodes), and
- the IBM POWER5+ system Huygens (sixteen 1.9-Ghz IBM Power5+ cores and 64 GB shared memory per node, Infiniband network with 1.2 GB/s bandwidth and $4.5\mu\text{s}$ latency between nodes).

For all experiments we take for the domain Ω the unit square, decompose Ω in p square subdomains, and cover each subdomain by a 256×256 grid according to §3.1. So the total number of unknowns depends linearly on p . (Except for the speed-up: there we fix the overall grid size and vary the subdomain grid size.) For the experiments that are presented after §5.1, the eigenvalue problem for the two-dimensional Laplace operator is considered. We start Jacobi-Davidson with a parabolic shaped vector¹ and we compute the global eigenvector of the operator on Ω until the residual norm of the approximate eigenpair is less than 10^{-9} . For the approximate solution of the correction equations (8) and (20) in the two processes, right preconditioning has been applied for exact solves with the preconditioner \mathbf{M} (i.e. exact subdomain solves) to enable a Schur complement approach. The preconditioner \mathbf{M} is constructed only once, at the first Jacobi-Davidson outer iteration.

On the linux-clusters DAS-2, DAS-3, and LISA, one subdomain is assigned to one node. On the IBM POWER5+ system Huygens, however, one subdomain is assigned to one core. Implementation is in Fortran77 with calls to BLAS, LAPACK, and MPI. To check the reliability, all experiments have been performed three times. Results presented here are averages of the three measured wall-clock times. For the DAS-2 linux-cluster the standard deviation of the three measured wall-clock times was always less than 1 %. For the other supercomputers the standard deviation σ was a bit larger, this is shown by the vertical lines (the interval $[T_{\text{average}} - \sigma, T_{\text{average}} + \sigma]$) in Fig. 6.

¹In practical applications, when Jacobi-Davidson has not arrived yet in the region of fast convergence (which is quadratic/cubic in case of exact solution of the correction equation) one should use the target value τ instead of the approximate eigenvalue θ as a shift in the correction equation (and the projections can be omitted). However, in most cases (for instance when moving onwards to compute the next eigenpair) Jacobi-Davidson is in this region. With this parabolic shaped startvector Jacobi-Davidson just enters the region of fast convergence (see [6, §3.5.1]).

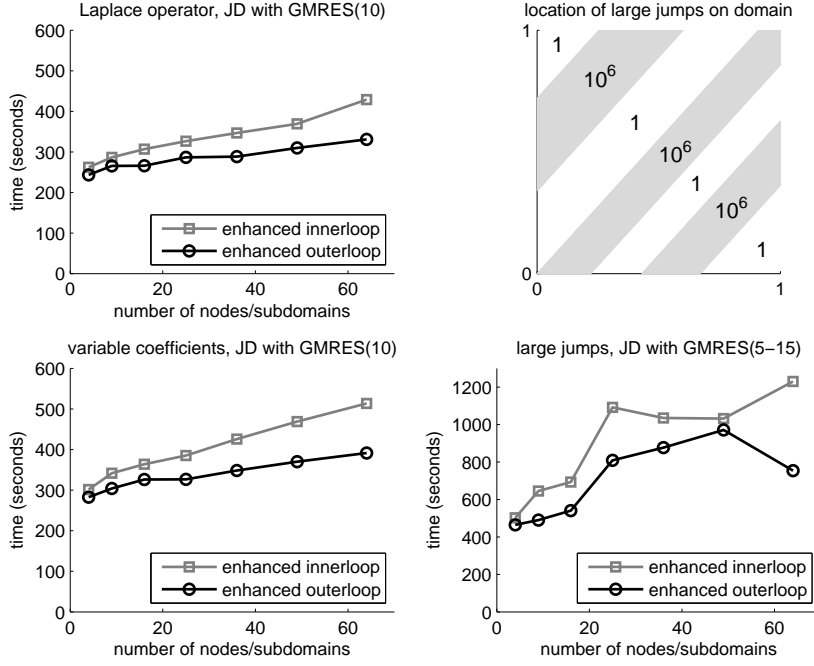


FIGURE 3. Parallel performance for different eigenvalue problems on DAS-2 linux-cluster.

5.1 Different eigenvalue problems

First we study the parallel performance for different two-dimensional eigenvalue problems. Considered are the eigenvalue problem of the Laplace operator:

$$\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2},$$

an advection-diffusion operator with variable (oscillatory) coefficients (taken from [1]):

$$\begin{aligned} \frac{\partial}{\partial x} \left[\left(1 + \frac{1}{2} \sin(50\pi x) \right) \frac{\partial}{\partial x} \right] + \frac{\partial}{\partial y} \left[\left(1 + \frac{1}{2} \sin(50\pi x) \sin(50\pi y) \right) \frac{\partial}{\partial y} \right] \\ - \left(20 \sin(10\pi x) \cos(10\pi y) \right) \frac{\partial}{\partial x} + \left(20 \cos(10\pi x) \sin(10\pi y) \right) \frac{\partial}{\partial y}, \end{aligned}$$

and a diffusion operator with large discontinuous jumps in the coefficients:

$$\frac{\partial}{\partial x} \left[D(x, y) \frac{\partial}{\partial x} \right] + \frac{\partial}{\partial y} \left[D(x, y) \frac{\partial}{\partial y} \right],$$

with $D(x, y) = 1$ or 10^6 , depending on location (see right top picture in Fig. 3). For the first two cases we take a fixed number of 10 GMRES inner iterations per Jacobi-Davidson outer iteration (see §5.2 for other choices). For the operator with large jumps a variable number between 5 and 15 of GMRES inner iterations has been taken, with the inexact Newton stopping criterion (21) from §5.2. These choices are based on practical observations and almost optimal.

For these three different eigenvalue problems, from the results in Fig. 3 one observes that Jacobi-Davidson with enhanced outerloop outperforms Jacobi-Davidson with enhanced innerloop.

For the advection-diffusion operator with variable coefficients the first order terms represent a 10×10 array of closed convection cells. Although for the decomposition in 2×2 and 5×5 subdomains (corresponding to 4 and 25 nodes, respectively) these cells align with the subdomain boundaries, no significant change in behavior can be observed in Fig. 3 (left bottom picture) compared to the other decompositions. Note that, in [1] this operator was shifted by a constant close to an interior eigenvalue, resulting in an indefinite linear system which is hard to solve. For Jacobi-Davidson we solve a similar linear system: the correction equation, however, here the conditioning is improved by projections/deflation (see also end of §3.2).

Although, for the diffusion operator with large jumps we choose the locations of the jumps such that they do not align with the subdomains, the geometry may be the reason for the somehow irregular behavior for more than 4×4 subdomains (16 nodes) in Fig. 3 (right bottom picture). But still Jacobi-Davidson with enhanced outerloop is faster.

5.2 Different stopping strategies

As already remarked, Jacobi-Davidson can be viewed as an accelerated inexact Newton method. Because of this the accuracy of the solution of the correction equation influences the convergence rate of the Jacobi-Davidson outer iterations. However, there is a trade off: exact solution will lead to a minimal number of outer iterations, but in general this will be more expensive (in terms of computational time) than approximate solves for some specific accuracy. For an efficient overall method, it was suggested in [4, §4] to use at the j^{th} Jacobi-Davidson outer iteration

$$\|\mathbf{r}^{(i)}\|_2 < 2^{-j} \|\mathbf{r}^{(0)}\|_2 \quad (21)$$

as a stopping criterion for the inner iterations. Here $\mathbf{r}^{(0)}$ is the residual at the start of the inner iterations and $\mathbf{r}^{(i)}$ the residual at the i^{th} inner iteration. In general, for inexact Newton methods such a stopping criterion may result in an efficient approach [2].

Here we investigate the inexact Newton stopping criterion (21) for Jacobi-Davidson with enhanced innerloop and enhanced outerloop. As, from the example §4.4.2 we already know that a relative low number of inner iterations is less beneficial for Jacobi-Davidson with enhanced innerloop, we consider the following approaches:

- GMRES(1 – 20) – a number of GMRES inner iterations that varies between 1 and 20 with (21) as stopping criterion,
- GMRES(5 – 15) – a number of GMRES inner iterations that varies between 5 and 15 with (21) as stopping criterion, and
- GMRES(10) – a fixed number of 10 GMRES inner iterations.

Results are presented in Fig. 4. The figure also shows the minimum, taken over the wall-clock times of the three approaches (right bottom picture). For these three different approaches, from the results one observes that Jacobi-Davidson with enhanced outerloop outperforms Jacobi-Davidson with enhanced innerloop.

Furthermore, a not that low fixed number of inner iterations (see GMRES(10) approach in left bottom picture of Fig. 4) is more beneficial for Jacobi-Davidson with enhanced innerloop. For the approaches with the inexact Newton stopping criterion, at the first Jacobi-Davidson outer iterations a low number of inner iterations have been taken. From §4.4 we know that, in such a situation, the error components due to the matrix splitting $\mathbf{M} - \mathbf{N}$ (i.e. due to neglecting the part \mathbf{N} that couples subdomains in first instance) are not damped that much and the next outer iteration this damping is disturbed for Jacobi-Davidson with

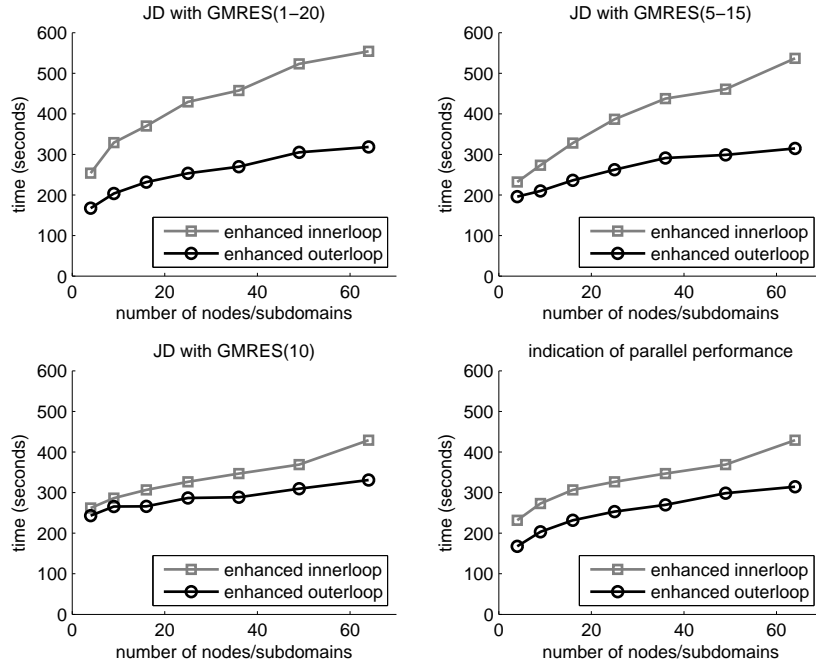


FIGURE 4. Parallel performance for different stopping strategies for the inner iteration on DAS-2 linux-cluster.

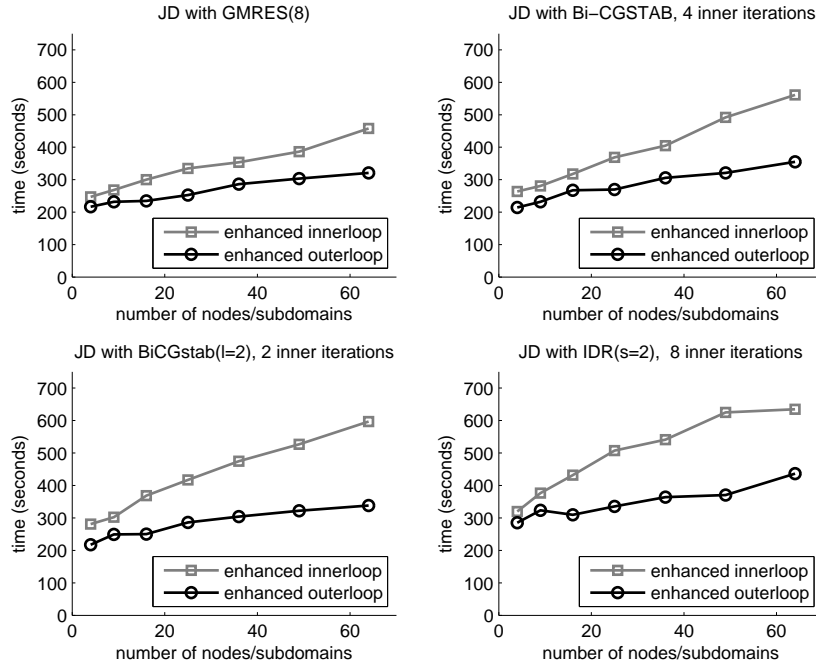


FIGURE 5. Parallel performance for different inner solvers on DAS-2 linux-cluster.

enhanced innerloop. Therefore, for convergence additional work has to be done afterwards, resulting in larger wall-clock times. From this we conclude that an inexact Newton stopping criterion does not suit very well for Jacobi-Davidson with enhanced innerloop.

However, Jacobi-Davidson with enhanced outerloop combines well with an inexact Newton stopping criterion. From Fig. 4 we see that for a low number of subdomains the approach with GMRES(10) is less efficient. Although in the enhanced outerloop error components from previous outer iterations are damped further in the next outer iteration, this may be explained by the number of error components. The number of error components depends on the number of subdomains, a low number requires effectively less GMRES iterations. For a larger number of subdomains the three approaches give similar results for Jacobi-Davidson with enhanced outerloop, therefore in the experiments that follow we shall use a fixed number of inner iterations as the basic approach.

5.3 Different inner solvers

Up till now we used GMRES as a Krylov method for the approximate solution of the correction equation. Here we compare the following Krylov methods:

- GMRES(8) [7],
- Bi-CGSTAB, 4 inner iterations [15],
- BiCGstab($\ell = 2$), 2 inner iterations [8], and
- IDR($s = 2$), 8 inner iterations [11].

The values of ℓ and s , and the numbers of inner iterations are chosen such that each outer iteration the number of matrix vector multiplications for the inner iterations is the same.

The aim is not to show which Krylov method is best. The point that we want to make here is that Jacobi-Davidson with enhanced outerloop profits more from the combination of the Jacobi-Davidson search subspace in the outerloop and the Krylov subspace in the innerloop, independent from the Krylov method that is used. This is confirmed by the results in Fig. 5.

Note that the total number of inner iterations is that small that it pays off to use GMRES (which searches for some optimal solution in the Krylov subspace). The three other Krylov methods are of interest when the number of inner iterations is larger and the Krylov subspace is getting too large for GMRES to handle (both computationally and in terms of memory usage). For our present experiments that is not the case.

Note also, that Jacobi-Davidson with enhanced innerloop suffers more from disturbing the damping for the three other Krylov methods than for GMRES. Jacobi-Davidson with enhanced outerloop is not that sensible for which Krylov method is used, some part of the larger wall-clock times for IDR(s) may be explained by the additional initial work for building the start vectors.

decomposition $p \times n^2$	enhanced	T_p (s)	$T_p \times p$	speed-up T_1/T_p	effectivity $T_1/(pT_p)$
1×65536	innerloop	$T_1 = 126.44$	126.44	-	-
1×65536	outerloop	126.94	126.94	-	-
4×16384	innerloop	24.91	99.64	5.08	1.27
4×16384	outerloop	17.79	71.16	7.11	1.78
16×4096	innerloop	3.75	60.06	33.69	2.11
16×4096	outerloop	2.81	44.93	45.03	2.81
64×1024	innerloop	0.88	56.01	144.48	2.26
64×1024	outerloop	0.68	43.21	187.29	2.93

TABLE 2. *Speed-up and effectivity on DAS-2 linux-cluster.*

5.4 Speed-up

Now we give an indication of the speed-up and effectivity which may be expected for Jacobi-Davidson with enhanced innerloop and enhanced outerloop. For this purpose we fix the overall grid size and vary the subdomain grid size. Table 2 shows the minimum of the wall-clock times for the three approaches GMRES(1 – 20), GMRES(5 – 15), and GMRES(10) of §5.2. In this table, T_1 is the wall-clock time of Jacobi-Davidson with exact solution of the enhanced correction equation on 1 node/domain (i.e. no decomposition in subdomains). T_p is the minimal wall-clock time of Jacobi-Davidson with approximate solution of the correction equation on p nodes/subdomains. Note that the amount of computational work for Jacobi-Davidson does not scale linearly with the number of grid points, therefore splitting the computational work in several smaller parts in a proper way (for instance with the domain decomposition technique) may really pay off (even on scalar computers). For parallel computers with distributed memory this is illustrated by Table 2.

5.5 Massively parallel behavior

In the right bottom picture of Fig. 4 the slope of the line for Jacobi-Davidson with enhanced outerloop is getting less when the number of subdomains increases. Question remains whether this nice trend holds on for a larger number of computational cores, i.e. whether Jacobi-Davidson with enhanced outerloop scales well for massively parallel computing.

Fig. 6 shows the parallel performance of Jacobi-Davidson with enhanced innerloop and enhanced outerloop on the three different supercomputers Lisa, DAS-3, and Huygens. In the top picture (up to 196 nodes) we only included wall-clock times for 64 nodes of the DAS-3 linux-cluster as the Lisa linux-cluster turned out to be faster. For a better comparison, results of the Lisa linux-cluster are also shown in the bottom picture (up to 1024 cores) of Fig. 6.

For the Lisa and DAS-3 linux-clusters, and up to 400 cores of the IBM POWER5+ system Huygens we used a fixed number of 10 GMRES inner iterations. At the IBM POWER5+ system Huygens, after 256 cores a fixed number of 20 GMRES inner iterations turned out to be more efficient. This can again be explained by the number of error components: this number depends on the number of subdomains, a larger number requires more GMRES iterations. We verified that the choice GMRES(20) was still nearly optimal for 1024 cores by also using GMRES(1 – 30) (i.e. the number of GMRES inner iterations varying between 1 and 30 with (21) as stopping criterion), this is shown by Fig. 7.

From the results in Fig. 6 one can conclude that for large scale eigenvalue problems it pays off to use Jacobi-Davidson with the enhanced outerloop instead of the enhanced innerloop and that Jacobi-Davidson with enhanced outerloop scales very well for massively parallel computing.

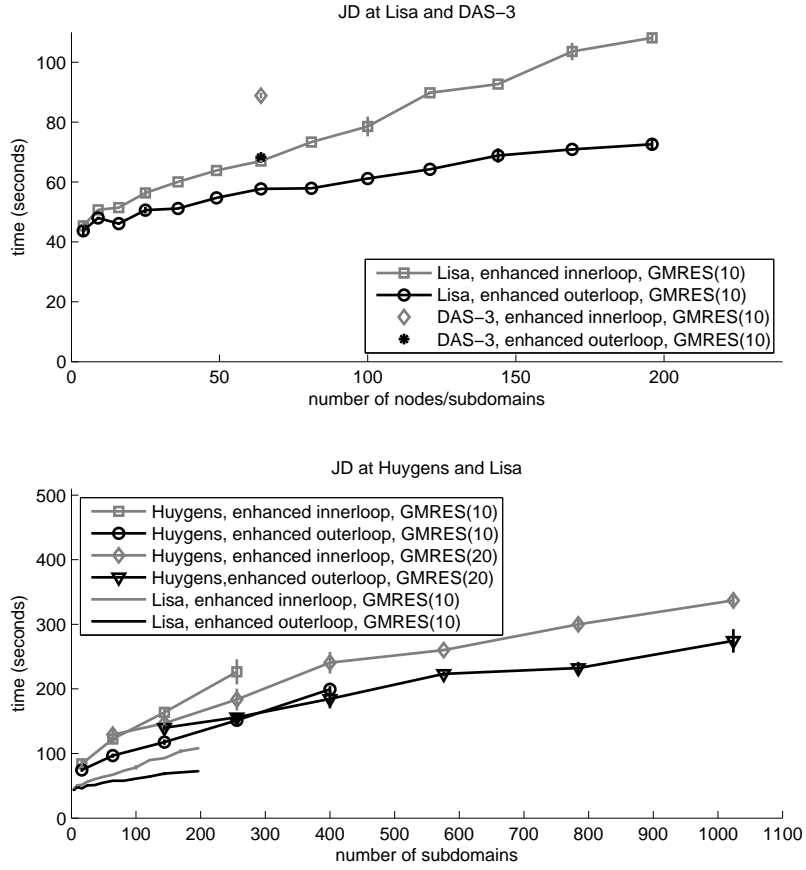


FIGURE 6. Massively parallel behavior: on Lisa and DAS-3 linux-clusters (top) and on IBM POWER5+ system Huygens and Lisa linux-cluster (bottom).

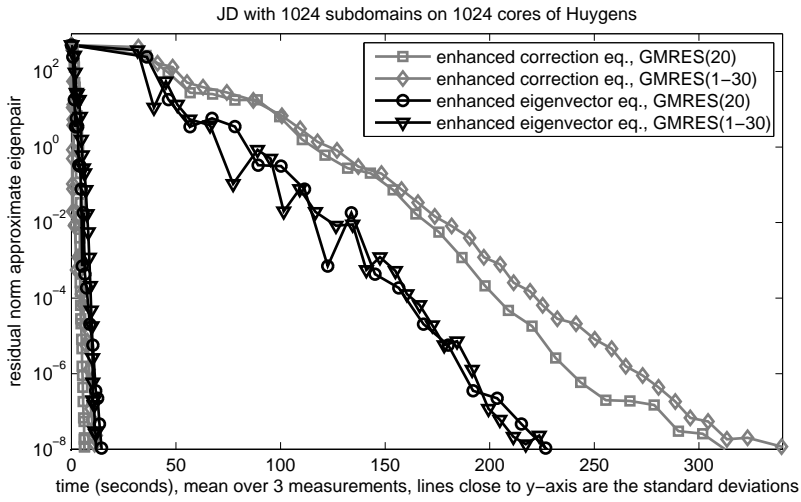


FIGURE 7. Convergence history for two different stopping strategies for the inner iteration on IBM POWER5+ system Huygens.

6 Conclusions

In this report two different levels of enhancement in the Jacobi-Davidson method have been considered for the computation of eigenvalues and eigenvectors of a matrix. These enhancements serve to incorporate a preconditioner based on domain decomposition in the correction equation of Jacobi-Davidson from previous research. By taking into account the relation of the intermediate level with the top level of Jacobi-Davidson, the previous approach has been improved.

For exact solutions of the correction equation the previous and the new approach are equivalent. But for approximate solutions of the correction equation that is not the case. This is because of the specific structure of the preconditioner: it is optimized for damping error components that originate from the matrix splitting based on domain decomposition. With an example it has been illustrated that for low accurate solutions of the correction equation the new approach should be preferred. For large scale eigenvalue problems that need a massively parallel treatment this aspect turned out to be nontrivial. The impact on the parallel performance has been shown by results of an extended set of scaling experiments on different supercomputers (up to 1024 cores) and by considering several aspects.

Results of the research presented here are also of interest for applications that use other type of inexact Newton methods for large systems of nonlinear equations.

Acknowledgements

The author wants to thank Gerard Sleijpen and Henk van der Vorst for the stimulating discussions at earlier stages of this research. Especially for the advice by Henk van der Vorst to consider one-step approximations (which resulted in §4.4.1) and writing out an example for this special case (which resulted in §4.4.2). Furthermore, the author wants to thank the Centrum Wiskunde & Informatica (CWI) in Amsterdam and the Physics Department of Utrecht University for providing computer facilities, ASCI/NWO-NCF for using the DAS-2 and DAS-3 linux-clusters, and SARA/NWO-NCF for using the Lisa linux-cluster and the IBM POWER5+ system Huygens.

This work was sponsored by the Stichting Nationale Computerfaciliteiten (National Computing Facilities Foundation, NCF) for the use of supercomputer facilities, with financial support from the Nederlandse Organisatie voor Wetenschappelijk Onderzoek (Netherlands Organization for Scientific Research, NWO).

References

- [1] X. -C. Cai, W. D. Gropp, and D. E. Keyes, A comparison of some domain decomposition and ILU preconditioned iterative methods for nonsymmetric elliptic problems, *Num. Lin. Alg. Appl.*, 1:477-504, 1994.
- [2] R. S. Dembo, S. C. Eisenstat, and T. Steihaug, Inexact Newton methods, *SIAM J. Numer. Anal.*, 19:400–408, 1982.
- [3] D. R. Fokkema, G. L. G. Sleijpen, and H. A. van der Vorst, Accelerated inexact Newton schemes for large systems of nonlinear equations, *SIAM J. Sci. Comput.*, 19:657–674, 1998.
- [4] D. R. Fokkema, G. L. G. Sleijpen, and H. A. van der Vorst, Jacobi-Davidson style QR and QZ algorithms for the reduction of matrix pencils, *SIAM J. Sci. Comput.*, 20:94–125, 1998.

- [5] M. Genseberger, G. L. G. Sleijpen and H. A. van der Vorst, Using domain decomposition in the Jacobi-Davidson method, Preprint 1164, Department of Mathematics, Utrecht University, 2000. Under revision for publication.
- [6] M. Genseberger, Domain decomposition in the Jacobi-Davidson method for eigenproblems, Chapter 3 and 4 of Ph.D. thesis, Utrecht University, The Netherlands, 2001.
- [7] Y. Saad and M. H. Schultz, GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, *SIAM J. Sci. Stat. Comp.*, 7:856–869, 1986.
- [8] G. L. G. Sleijpen and D. R. Fokkema, BiCGstab(ℓ) for linear equations involving matrices with complex spectrum, *Electron. Trans. Numer. Anal.*, 1:11–32, 1993.
- [9] G. L. G. Sleijpen and H. A. van der Vorst, A Jacobi-Davidson iteration method for linear eigenvalue problems, *SIAM J. Matrix Anal. Appl.*, 17:401–425, 1996.
- [10] G. L. G. Sleijpen, H. A. van der Vorst, and E. Meijerink, Efficient expansion of subspaces in the Jacobi-Davidson method for standard and generalized eigenvalue problems, *Electron. Trans. Numer. Anal.*, 7:75–89, 1998.
- [11] P. Sonneveld and M. B. van Gijzen, IDR(s): a family of simple and fast algorithms for solving large nonsymmetric linear systems, Technical Report 07-07, Department of Applied Mathematical Analysis, Delft University of Technology, Delft, The Netherlands, 2007.
- [12] K. H. Tan and M. J. A. Borsboom, On generalized Schwarz coupling applied to advection-dominated problems, Domain decomposition methods in scientific and engineering computing (DD7, University Park, PA, 1993), *Amer. Math. Soc.*, Providence, RI, 125–130, 1994.
- [13] K. H. Tan, Local Coupling in Domain Decomposition, Ph.D. thesis, Utrecht University, The Netherlands, 1995.
- [14] W. P. Tang, Generalized Schwarz Splittings, *SIAM J. Sci. Stat. Comp.*, 13:573–595, 1992.
- [15] H. A. van der Vorst, Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems, *SIAM J. Sci. Stat. Comp.*, 13:631–644, 1992.